I Erläuterungen

Voraussetzungen gemäß KCGO und Abiturerlass in der für den Abiturjahrgang geltenden Fassung

Standardbezug
Die nachfolgend ausgewiesenen prozessbezogenen Kompetenzbereiche sind für die Bearbeitung der jeweiligen Aufgabe besonders bedeutsam. Darüber hinaus können weitere, hier nicht ausgewiesene prozessbezogene Kompetenzbereiche für die Bearbeitung der Aufgabe nachrangig bedeutsam sein, zumal die Kompetenzbereiche in engem Bezug zueinander stehen. Die Operationalisierung des Standardbezugs erfolgt in Abschnitt II.

Aufgabe	Prozessbezogene Kompetenzbereiche				
	P1	P2	P3	P4	P5
1		X			
2			X		
3			X		
4		X	X		
5		X		X	
6		X		X	

Inhaltlicher Bezug
Der vorliegende Vorschlag bezieht sich schwerpunktmäßig auf die inhaltsbezogenen Kompetenzbereiche Algorithmen (I1), Information und Daten (I3) nach KCGO.

Q1: Algorithmik und objektorientierte Modellierung
verbindliche Themenfelder: Such- und Sortieralgorithmen (Q1.1); Rekursion (Q1.2); Klassen und Objekte (Q1.3); Höhere Datenstrukturen und ihre objektorientierte Modellierung (Q1.4)

II Lösungshinweise und Bewertungsraster

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, sind ebenso zu akzeptieren.

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE
1	Da bei der neuen Software beim Zusammenstellen eines Wanderweges noch nicht bekannt ist, aus wie vielen Abschnitten die Wanderung besteht, ist es sinnvoll eine Datenstruktur zu wählen, die dynamisch leicht änderbar ist. Dies gilt für die Datenstruktur Liste, da jederzeit Abschnitte hinzugefügt bzw. gelöscht werden können. Bei der Datenstruktur Feld muss hingegen die Länge festgelegt werden und zudem sind Einfüge- und Löschoperationen mit aufwändigen Verschiebungen von Stationen verbunden.	4

2

```
TWanderweg
- Name: string
- ersterAbschnitt: TAbschnitt
c create()
+ erzeugeAbschnitte(stationen: TStationenArray)
+ ermittleRundweg()
+ sucheRundweg(einAbschnitt: TAbschnitt): boolean
+ berechneGesamtdauer(): double
```

```
TAbschnitt
- Laenge: double
- Schwierigkeitsgrad: integer
- Beginn: TStation
- Ende: TStation
- naechsterAbschnitt: TAbschnitt
c create(Beginn: TStation, Ende: TStation)
+ berechneDauer(): double
```

```
TStation
- Name: string
- Schutzhuette: boolean
c create(name: string, schutzhuette: boolean)
```

Hinweis
Die Methode *sucheRundweg(...)* kann auch als *private (-)* modelliert werden.

BE: 6

3

```
procedure TWanderweg.erzeugeAbschnitte(Stationen: TStationenArray);
var i: integer;
    laufAbschnitt, neu: TAbschnitt;
begin
  ersterAbschnitt:= TAbschnitt.create(Stationen[0], Stationen[1]);
  laufAbschnitt:= ersterAbschnitt;
  i:= 1;
  while i < Length(Stationen) - 1 do begin
    neu:= TAbschnitt.create(Stationen[i], Stationen[i+1]);
    laufAbschnitt.setNaechsterAbschnitt(neu);
    laufAbschnitt:= neu;
    i:= i+1;
  end;
end;
```

BE: 5

4.1

```
function TAbschnitt.berechneDauer: double;
begin
  result:= Laenge/(4.2-0.5*(Schwierigkeitsgrad-1));
end;
```

BE: 2

4.2

```
Algorithmus berechneGesamtdauer()
Dauer = 0
einAbschnitt = ersterAbschnitt
wiederhole solange einAbschnitt <> nil
    Dauer = Dauer + einAbschnitt.berechneDauer()
    einAbschnitt = einAbschnitt.getNaechsterAbschnitt()
Rückgabe Dauer
```

BE: 2

Aufg.	erwartete Leistungen	BE
5	Die Prozedur *ermittleRundweg* wird ohne Parameter aufgerufen. Sie gibt auf der Konsole die Meldung "Kein Rundweg gefunden!" aus, wenn die Funktion *sucheRundweg(...)* den Rückgabewert *false* liefert. Die Funktion *sucheRundweg(...)* wird mit dem Parameter *einAbschnitt* vom Typ *TAbschnitt* aufgerufen und liefert einen booleschen Rückgabewert. In Zeile 8 wird die boolesche Variable *gefunden* deklariert und in Zeile 12 mit *false* initialisiert. Sofern *einAbschnitt* einen Nachfolger besitzt, wird in Zeile 14 ermittelt, ob dieser der Beginn eines Rundweges ist. Dazu wird *sucheRundweg(...)* rekursiv mit dem entsprechenden Parameter aufgerufen und der Rückgabewert der Variablen *gefunden* zugewiesen. In den Zeilen 15 bis 24 wird untersucht, ob *einAbschnitt* der Beginn eines Rundweges ist. Dazu wird die Variable *laufAbschnitt* mit dem Wert von *einAbschnitt* und die Variable *Beginn* mit dem Namen der Beginnstation initialisiert. Die folgende while-Schleife wird solange durchlaufen wie *laufAbschnitt* nicht *nil* ist. In dieser Schleife wird geprüft, ob *laufAbschnitt* das Ende eines Rundweges ist, der mit *einAbschnitt* beginnt. Dazu wird verglichen, ob der Name der Endstation von *laufAbschnitt* mit dem Namen der Beginnstation von *einAbschnitt* übereinstimmt. In diesem Fall wird dieser Name ausgegeben und die Variable *gefunden* auf *true* gesetzt. Um den nächsten Abschnitt zu vergleichen, wird *laufAbschnitt* auf den nächsten Abschnitt gesetzt. In Zeile 25 erfolgt die Rückgabe, ob ein Rundweg gefunden wurde. Durch den rekursiven Aufruf von *sucheRundweg(...)* in Zeile 14 werden alle vorhandenen Rundwege gefunden und ausgegeben.	9
6.1	Nach Entfernung des Rundweges besteht der Wanderweg nur noch aus der Abschnittsfolge A-G. Station 2 ist Anfang und Ende des Rundweges, somit sind alle Abschnitte dazwischen zu entfernen. Auf den Abschnitt A folgt nun als nächster der Abschnitt G.	3
6.2	Wird ein Rundweg gefunden, wird das Attribut *naechsterAbschnitt* des letzten Abschnitts vor dem gefundenen Rundweg auf den Abschnitt gesetzt, der auf den gefundenen Rundweg folgt. Befindet sich ein Rundweg am Beginn des Wanderweges, wird das Attribut *ErsterAbschnitt* des Wanderweges auf den ersten Abschnitt nach dem Rundweg gesetzt.	4
	Summe	**35**

III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) in der jeweils geltenden Fassung. Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 Satz 3 OAVO in Verbindung mit Anlage 9b anzuwenden.

Bei der Bewertung und Beurteilung der Übersetzungsleistung in den Fächern Latein und Altgriechisch sind die Bestimmungen des § 9 Abs. 14 OAVO in Verbindung mit Anlage 9c anzuwenden.

Der Fehlerindex ist nach Anlage 9b zu § 9 Abs. 12 OAVO zu berechnen. Für die Ermittlung der Punkte nach Anlage 9a zu § 9 Abs. 12 OAVO sowie Anlage 9c zu § 9 Abs. 14 OAVO wird jeweils der ganzzahlige nicht gerundete Prozentsatz bzw. Fehlerindex zugrunde gelegt.

Für die Bewertung in den modernen Fremdsprachen ist der „Erlass zur Bewertung und Beurteilung von schriftlichen Arbeiten in allen Grund- und Leistungskursen der neu beginnenden und fortgeführten modernen Fremdsprachen in der gymnasialen Oberstufe, dem beruflichen Gymnasium, dem Abendgymnasium und dem Hessenkolleg“ vom 7. August 2020 (ABl. S. 519) zugrunde zu legen. Demnach erfolgt die Bewertung und Beurteilung mit der Maßgabe, dass lediglich bei der Ermittlung des Prüfungsergebnisses (Note) aus Prüfungsteil 1 und 2 gerundet wird.

Darüber hinaus sind die Vorgaben der Erlasse „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen (Abiturerlass)“ und „Durchführungsbestimmungen zum Landesabitur“ in der für den Abiturjahrgang geltenden Fassung zu beachten.

Im Fach Informatik (Leistungskurs) werden Vorschläge zu den Themen der drei Kurshalbjahre Q1 (Algorithmik und objektorientierte Modellierung), Q2 (Datenbanken) und Q3 (Konzepte und Anwendungen der theoretischen Informatik) vorgelegt, wobei die Prüfungsleistung aus der Bearbeitung je eines Vorschlags zu jedem Halbjahresthema besteht, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten (ausreichend)** setzt voraus, dass mindestens 45% der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass mindestens 75% der zu vergebenden BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	**Bewertungseinheiten in den Anforderungsbereichen**			**Summe**
	AFB I	**AFB II**	**AFB III**	
1	2	2		**4**
2	2	4		**6**
3		4	1	**5**
4	2	2		**4**
5	3	2	4	**9**
6	1	4	2	**7**
Summe	**10**	**18**	**7**	**35**

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.